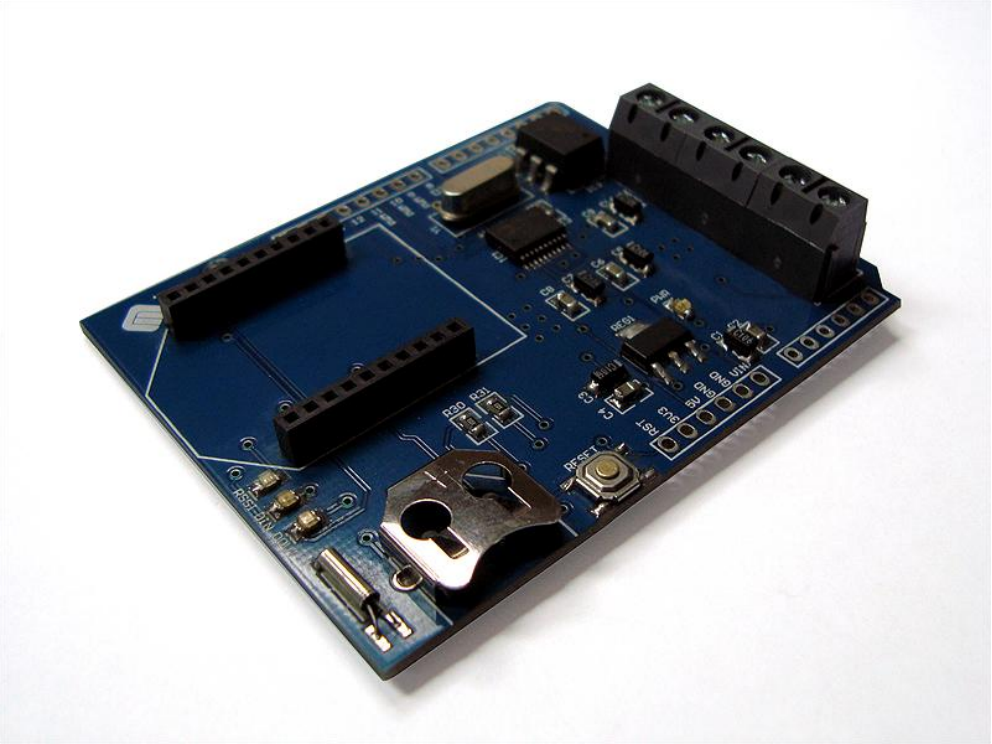
Manual de Usuario

Arduino Energy Meter Shield

Rev. A

MCI-MA-0164



INGENIERÍA MCI LTDA.

www.olimex.cl

Luis Thayer Ojeda 0115. Of. 402 ▪ Santiago, Chile ▪ Tel. +56 2 3339579 ▪ info@olimex.cl

MCI Ltda.

Luis Thayer Ojeda 0115. Oficina 402
Santiago, Chile

www.olimex.cl

Tel: +56 2 3339579
Fax: +56 2 3350589

® MCI Ltda. 2012

Atención: Cambios y modificaciones hechas en el dispositivo, no autorizados expresamente por MCI, anularán su garantía. MCI no se hace responsable por daños a personas y/o sus propiedades por el uso inadecuado de este producto.

Código Manual: MCI – MA - 0164

CONTENIDO

CONTENIDO 3
INTRODUCCIÓN 4
CARACTERÍSTICAS GENERALES 5
DEFINICIONES 5
PARTES DEL SHIELD 6
INSTALACIÓN DEL SHIELD 8
MAPA DE PUERTOS 10
CARACTERÍSTICAS ELÉCTRICAS 10
CARACTERÍSTICAS MECÁNICAS 10
CONFIGURACIÓN RELOJ DE TIEMPO REAL 11
MIDIENDO ENERGÍA 12
NOTAS 20
MANTENIMIENTO 21
HISTORIA DEL DOCUMENTO 22

INTRODUCCIÓN

Arduino Energy Shield es una tarjeta de que permite monitorear el consumo de energía de tu casa, oficina, instalación o equipo directamente con tu tarjeta Arduino, sin necesidad de cables usando la comunicación inalámbrica XBee. Arduino Energy Shield incorpora el chip ADE7753 que permite medir la potencia activa, potencia reactiva, voltaje y corriente RMS, entre otras variables en un instante dado. Arduino Energy Shield es compatible con placas Arduino Duemilanove, Uno y Mega.

Arduino Energy Shield está pensado para ser utilizado por profesionales y hobistas que necesiten realizar pruebas y prototipos que contemplen la medición de variables eléctricas. No se recomienda su uso en dispositivos dedicados a tarificación eléctrica.

Arduino Energy Shield cuenta con un reloj de tiempo real que permite tomar la hora y fecha en que se tomó el dato y transmitirlo a otro dispositivo para su procesamiento y visualización. También cuenta con un socket para baterías tipo coin de 12mm, como alimentación de respaldo del reloj de tiempo real.

Para transmitir los datos a otro equipo la Arduino Energy Shield viene con un socket compatible con módulos XBee y XBee Pro, eliminando la necesidad de cablear desde el punto de medición hasta el equipo donde se guardan, procesan y visualizan los datos capturados por la Arduino Energy Shield.

El Arduino Energy Shield viene con completas bibliotecas de funciones, útiles para realizar todo tipo de operaciones con el medidor, desde tareas de calibración hasta obtención de datos capturados. Además se incluyen librerías con funciones para comunicarse con el reloj de tiempo real incluido.

CARACTERÍSTICAS GENERALES

- Medidor de potencia activa y aparente
- Medidor de Voltaje y corriente RMS
- Rango de operación entre 0[V] y 220[V]
- Calibrado para corriente máxima de 30[A]
- Sensor de temperatura capaz de medir entre -25°C y 80°C con 3°C de precisión.
- Entrada sensores
  - Entrada de Voltaje: 12VAC
  - Entrada de Corriente: 20mA máximo.
- Salida de pulsos optoacoplada, de frecuencia proporcional a watt-hora medidos.
- Socket Batería Tipo Coin 12mm
- Reloj de tiempo real
  - IC DS1307
  - Reloj de 32kHz
- Botón de reset Arduino
- Alimentación 5 V_{DC}
- Socket para conexión de módulo XBee
- Conectores para sondas de medición aisladas galvánicamente.

DEFINICIONES

RSSI: *Receive Signal Strength Indication*, Indicador de fuerza de señal de recepción.

XBee: Módulo que permite la comunicación inalámbrica entre dispositivos usando la banda de frecuencias de 2.4GHz Soporta redes punto a punto, punto a multipunto y mesh.

Reloj de tiempo real (RTC, real time clock): Circuito integrado que mide el tiempo de manera exacta. En el caso de Arduino Energy Shield, este módulo entrega tanto la hora como la fecha, entre otras características a las que puede accederse mediante la librería que es provista.

Salida de pulsos: Los antiguos medidores análogos poseían una rueda giratoria, cierto número de giros representaba que se ha medido 1 KWh. En Arduino Energy Shield esta útil función sigue vigente gracias a la inclusión de esta salida de pulsos, que se encuentra conectada ópticamente a la circuitería interna.

PARTES DEL SHIELD

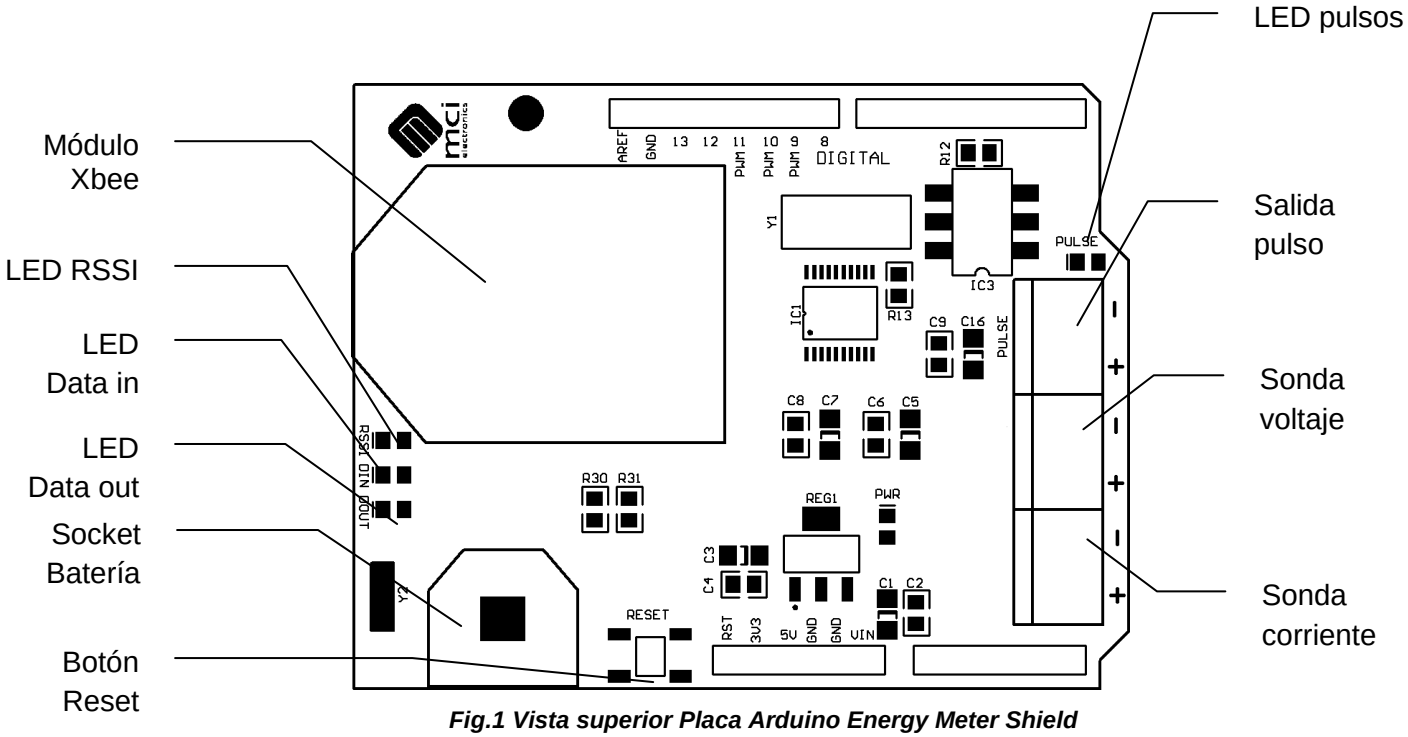


Fig.1 Vista superior Placa Arduino Energy Meter Shield

- **Módulo Xbee:** Socket de módulo Xbee
- **LED RSSI:** Led indicador de la potencia de la señal recibida por el módulo Xbee.
- **LED Data in:** LED que indica la transferencia de datos *hacia* el módulo XBee, si se encuentra encendido significa que se están transfiriendo datos al módulo.
- **LED Data out:** LED que indica la transferencia de datos *desde* el módulo XBee, si se encuentra encendido significa que se están transfiriendo datos desde módulo.
- **Socket batería:** Espacio para batería de respaldo de reloj de tiempo real.
- **Reset:** Botón para resetear la placa Arduino.
- **LED pulsos:** LED que parpadea a determinada frecuencia en función del consumo, esta frecuencia depende de la constante fijada al medidor para tal efecto, la salida de Arduino Energy Meter Shield entrega 32.000 impulsos/kWh.
- **LED PWR*:*** Led indicador de que placa se encuentra energizada

  - **Encendido:** Placa energizada
  - **Apagado:** Placa desenergizada
- **Salida pulsos:** Salida optoacoplada a la señal del LED de pulsos, puede ser utilizada para calibrar el equipo así como para labores de monitoreo.
- **Sonda voltaje:** Terminal de conexión del transformador medidor de voltaje 12 VAC.
- **Sonda corriente:** Terminal donde se conecta un sensor de corriente AC 20mA máx. no invasivo que funciona midiendo el campo magnético generado por la corriente que circula por el conductor.

INSTALACIÓN DEL SHIELD

Para conectar el Arduino Energy Meter Shield a la placa base Duemilanove, Mega o Uno se deben realizar los siguientes pasos:

A. Conectar el módulo Xbee al Shield, notar en la serigrafía del shield la orientación que debe tener el módulo XBee.
B. Insertar pila de Litio en socket correspondiente.
C. Conectar la sonda de corriente a terminales correspondientes.
D. Conectar la sonda de voltaje a terminales correspondientes.
E. Posicionar el shield en la forma que se muestra en la figura 2. Notar que los pin headers que bajan hacia la placa base Arduino tienen una única posición de conexión.
F. Ensamblar Arduino Energy Meter Shield con placa Arduino Base
G. Conectar Placa Base Arduino con PC usando cable USB

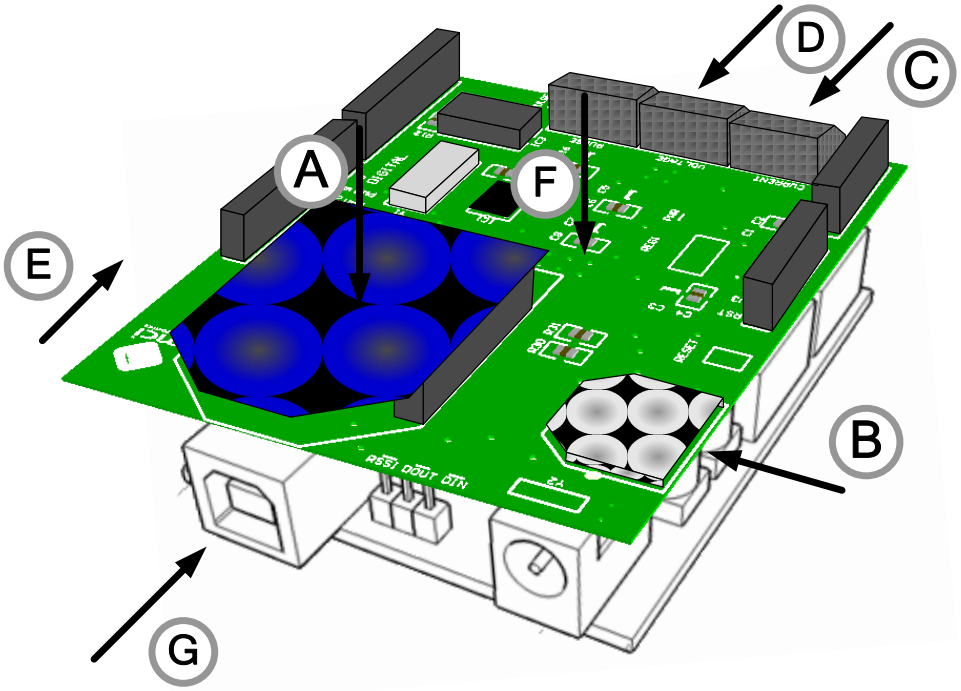


Fig.2 Ensamblaje Arduino Energy Meter Shield a placa Arduino

MAPA DE PUERTOS

Los puertos utilizados por el Arduino Energy Meter Shield no pueden ser utilizados por otro Shield, excepto por las señales SDA, SCL y RESET.

Pin	Nombre	Función
DIGITAL 0	Serial RX	Conexión a Rx del puerto serial por hardware. No utilizado
DIGITAL 1	Serial TX	Conexión a Tx del puerto serial por hardware. No utilizado
DIGITAL 2	SoftSerial RX	Conexión a Rx del puerto serial por software
DIGITAL 3	SoftSerial TX	Conexión a Tx del puerto serial por software
DIGITAL 10	CHIP SELECT	Para habilitar comunicación con medidor
DIGITAL 11	DATA IN	Datos de entrada en comunicación con medidor
DIGITAL 12	DATA OUT	Datos de salida en comunicación con medidor
DIGITAL 13	SIGNAL CLOCK	Señal de reloj en comunicación con medidor
ANALOG 4	I2C RTC SDA	Señal de datos comunicación I2C con Reloj
ANALOG 5	I2C RTC SCL	Señal de reloj comunicación I2C con Reloj

CARACTERÍSTICAS ELÉCTRICAS

- Voltaje de alimentación de 5 V_{DC}
- Socket Batería Tipo Coin 12mm 3V
- Consumo 5.4mA sin módulo XBee, 52.4mA con módulo XBee conectado.

CARACTERÍSTICAS MECÁNICAS

Dimensiones (Ancho x Largo x Alto) 54x69x12 [mm]

CONFIGURACIÓN RELOJ DE TIEMPO REAL

Para configurar el reloj de tiempo real de Arduino Energy Meter Shield se usa el programa para Arduino **Configure.pde** descargable desde el sitio web del producto en http://www.olimex.cl/, que al ser cargado en la placa Arduino permite configurar el reloj de tiempo real.

Al ejecutar el sketch **Configure** se preguntará primero por la fecha actual y luego por la hora. Tras esto se configurará el reloj usando los parámetros entregados. Finalmente quedará corriendo el reloj mostrando la fecha y hora actual.

A continuación se muestra un ejemplo del texto observado en la consola serial de Arduino al ejecutar el sketch.

```
+---------------------------+
| PROGRAMA DE CONFIGURACION |
|    ENERGY METER SHIELD    |
+---------------------------+
|  MCI Electronics - 2011   |
+---------------------------+


Configuracion de reloj
Ingrese fecha en formato dd/mm/aa
Ingrese hora en formato hh:mm:ss
Reloj configurado con exito
09/02/2011 - 17:25:40
09/02/2011 - 17:25:41
09/02/2011 - 17:25:42
09/02/2011 - 17:25:43
09/02/2011 - 17:25:44
09/02/2011 - 17:25:45
09/02/2011 - 17:25:46
```

MIDIENDO ENERGÍA

Debido al circuito integrado (CI) ADE7753 que se encuentra en el Arduino Energy Shield, este es capaz de realizar mediciones de Voltaje RMS, Corriente RMS, Energía Activa acumulada, Energía Reactiva Acumulada, Temperatura y Período de la señal de voltaje.

Los valores entregados por el CI ADE7753 son palabras de un largo de bits variable, dependiendo lo que se esté midiendo, estos valores deben ser transformados al valor real de la magnitud en estudio, dicha transformación se efectúa mediante una constante de proporcionalidad.

Voltaje RMS: Se tiene que la señal de voltaje debe ser conectada al canal 2 del Arduino Energy Shield, luego debido a que el CI ADE7753 debe tener una entrada de a lo más 1 Volt Peak to Peak, los dispositivos presentes en el Shield producen una atenuación de la señal en proporción 51:1.

El CI ADE7753 posee un modo de sincronización entre el momento de medición del voltaje RMS y los cruces por cero de la señal de voltaje, tomando esto en consideración y realizando una serie de 100 muestras para luego tomar un promedio se logra un resultado bastante estable. El resultado anterior se encuentra codificado en una palabra de 24 bits, y es proporcional al valor real.

Debido a la proporcionalidad existente entre la palabra de 24 bits que entrega el ADE7753 y el valor real del Voltaje RMS, es necesario encontrar la constante de proporcionalidad que permita realizar la transformación, dicha constante es propia del Shield.

Para encontrar la constante de proporcionalidad se puede proceder de la siguiente forma:

Se conecta al Shield una entrada con un Voltaje RMS conocido Vin, luego el valor que entra al CI ADE7753 corresponde a Vin /51 debido a la atenuación de la señal previamente mencionada. Finalmente la constante de proporcionalidad corresponderá a Vin/51 dividido por el valor mostrado por el CI ADE7753.

Corriente RMS: Se tiene que la señal de corriente debe ser conectada al canal 1 del Arduino Energy Shield, luego debido a que el CI ADE7753 debe tener una entrada de a lo más 1 Volt Peak to Peak, los dispositivos presentes en el Shield producen una atenuación de la señal en proporción 51:1.

El CI ADE7753 posee un modo de sincronización entre el momento de medición del corriente RMS y los cruces por cero de la señal de voltaje (Zero Crossing), tomando esto en consideración y realizando una serie de 100 muestras para luego tomar un promedio se logra un resultado bastante estable. El resultado anterior se encuentra codificado en una palabra de 24 bits, y es proporcional al valor real.

Debido a la proporcionalidad existente entre la palabra de 24 bits que entrega el ADE7753 y el valor real de la corriente RMS, es necesario encontrar la constante de proporcionalidad que permita realizar la transformación, dicha constante es propia del Shield.

Para encontrar la constante de proporcionalidad se puede proceder de la siguiente forma:

Se conecta al Shield una entrada con una Corriente RMS conocida Iin, luego el valor que entra al CI ADE7753 corresponde a Iin /51 debido a la atenuación de la señal, previamente mencionada. Finalmente la constante de proporcionalidad corresponderá a Iin/51 dividido por el valor mostrado por el CI ADE7753.

Energía Activa Acumulada: Se tiene que al conectar entradas de corriente y voltaje en el Shield en los canales 1 y 2, el CI ADE7753 acumula la energía activa desde que el Shield comienza a funcionar, en una palabra de 48 bits, sin embargo solo se puede tener acceso a los primeros 24 bits.

Si es que se quiere medir algún consumo, debido a este enfoque de acumulación que posee el CI ADE7753, es necesario tomar medidas de la energía acumulada en dos instantes de tiempo distintos, para luego calcular la diferencia entre estos. Se tiene que el CI ADE7737 realiza mediciones con una frecuencia aproximada de 4 MHz, por lo que debido a esta alta taza de muestreo es necesario un método especial para poder tener instantes de tiempo de medición precisos. Para solucionar el problema antes planteado el CI ADE7737 posee un modo de acumulación de energía durante un número de medios ciclos determinado (Cycmode), cada medio ciclo corresponde a la mitad de un ciclo de la señal de voltaje, por lo que teniendo la frecuencia de esta señal es posible calcular el tiempo que transcurre en la cantidad de medios ciclos previamente establecida, a modo de ejemplo, para la red eléctrica en Chile la frecuencia es de 50 Hz por lo que 100 medios ciclos equivalen a 1 segundo.

Se tiene que el valor entregado por la diferencia de energías es proporcional al valor real, y también tiene asociado un factor de 51, con esto la constante de proporcionalidad puede ser calculada de la siguiente forma:

Se conecta una entrada de corriente RMS conocida Iin en el canal 1 del Shield y una entrada de voltaje RMS conocida Vin en el canal 2, luego se calcula la Energía Activa acumulada. Para este ejemplo el tiempo de acumulación es de un segundo, con lo que el valor corresponde a la multiplicación entre Iin y Vin, luego el valor de salida del CI ADE7753 está relacionado a la multiplicación entre Iin y Vin (en el caso de una carga activa) dividida posteriormente por 51. Finalmente las constante de proporcionalidad corresponde al valor ((Iin*Vin) /51) dividido por el valor entregado por el CI ADE7753.

A continuación se muestra un ejemplo para el cálculo de las constantes de proporcionalidad realizado en el sketch p_manual.ino:

Se ocupó una entrada de 9 Volts RMS en el canal 2, una entrada de 0.454 Amperes en el canal 1, produciéndose un consumo de 4.086 [Joules/seg].

```
---Voltaje RMS---
Constante de Proporcionalidad Kv: 0.000000172090864
Voltaje RMS a la salida del ADE7753 (24 bits): 1026174
Voltaje RMS a la salida del Energy Shield (x 51): 52334874
Voltaje RMS a la salida del Energy Shield [Volts]: 9.0063533782

---Corriente RMS---
Constante de Proporcionalidad Ki: 0.000000383094263
Corriente RMS a la salida del ADE7753 (24 bits): 23238
Corriente RMS a la salida del Energy Shield (x 51): 1185138
Corriente RMS a la salida del Energy Shield [Amperes]: 0.4540195465

---Energia Acumulada en 1 segundo---
Constante de Proporcionalidad Ke: 0.000404634523391
Energia Acumulada en 1 seg, a la salida del ADE7753 (24 bits): 198
Energia Acumulada en 1 seg, a la salida del Energy Shield (x 51): 10098
Energia Acumulada en 1 seg, a la salida del Energy Shield [Joules/seg]: 4.0859999656
```

Después de instalar el Arduino Energy Meter Shield, fijas las constantes de proporcionalidad y configurar el reloj de tiempo real, como se explica en las secciones anteriores, es posible comenzar a medir la energía. En primer lugar se debe cargar en la placa Arduino el *sketch* **demo.ino**, que incluye todo lo necesario para medir temperatura, corriente, voltaje, energía activa, reactiva y además leer la hora actual desde el reloj de tiempo real para luego enviar toda esta información a través de XBee y además por el puerto serial por Hardware de la placa Arduino. Sin embargo los valores entregados se encuentran codificados en palabras de un número de bits que varía según la magnitud que se esté midiendo.

A continuación se muestra el resultado de correr el sketch **demo** midiendo una carga que consume alrededor de 500[mA] a 220[V], los resultados se encuentran en un valor digital, deben ser multiplicados por la constante de proporcionalidad.

```
=====11/01/2011 - 16:29:35=====

Fecha: 11/01/2011

Hora: 16:29:35

Hora unix: 1294777775

voltaje [V]: 1401965 | 1403679 | 221 | 216 | 218 | 220

corriente [mA]: 125528 | 125310 | 496 | 496 | 496 | 495

Consumo: 95

Consumo aparente: 101

Energia act: 95 | 100

Energia apa: 2860 | 2861

Temperatura: 22 | 22 | 22 | 23

mode: 8
```

Por otro lado se tiene el programa **demo2.ino** que permite medir el valor real de voltaje RMS, corriente RMS y energía activa acumulada presentes en el Energy Shield, todo lo anterior con un 3% de error.

```
------Nueva Medicion------
29/02/2012 - 11:43:59
Voltaje RMS [V]: 8.993028640747070
Corriente RMS [A]: 0.463996315002441
Energia Activa Acumulada [J] en 1 seg : 4.219573020935058
Periodo de la senial de voltaje [seg]: 0.02197

------Nueva Medicion------
29/02/2012 - 11:44:08
Voltaje RMS [V]: 8.999689102172851
Corriente RMS [A]: 0.464215755462646
Energia Activa Acumulada [J] en 1 seg : 4.219573020935058
Periodo de la senial de voltaje [seg]: 0.02199

------Nueva Medicion------
29/02/2012 - 11:44:16
Voltaje RMS [V]: 8.999119758605957
Corriente RMS [A]: 0.463956403732299
Energia Activa Acumulada [J] en 1 seg : 4.219573020935058
Periodo de la senial de voltaje [seg]: 0.02200
```

El proyecto cuenta también con la biblioteca de funciones para el manejo del chip ADE7753, desarrollada por MCI Electronics.

Para que la comunicación entre módulos XBee sea efectiva, éstos deben estar configurados correctamente para comunicarse unos con otros.

Recibiendo Datos desde Arduino Energy Shield

Para recibir los datos, puede usarse un XBee Explorer conectado a un computador, en el cual debe obtenerse la información como si de un puerto serial se tratara. Lo anterior puede hacerse usando el programa Hyperterminal como se ejemplifica en la sección siguiente.

Los datos recibidos vía XBee tienen el siguiente formato:

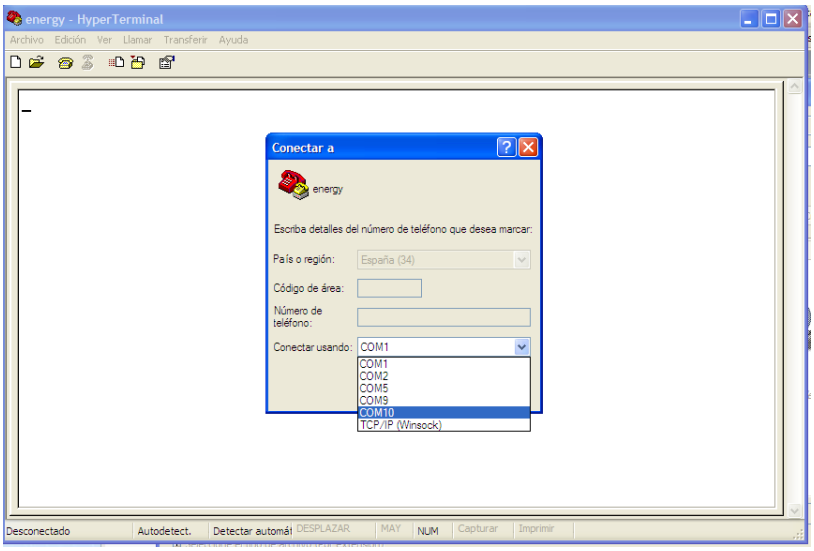
ID medidor	Tiempo [s]	Temperatura [°C]	Voltaje [V]	Corriente [mA]	Energía activa [Wh]	Consumo [W/3600]	Energía aparente [VAh]

- Tiempo está en formato UNIX Time, correspondiente a los segundos transcurridos desde las 0:00:00 1/1/1970

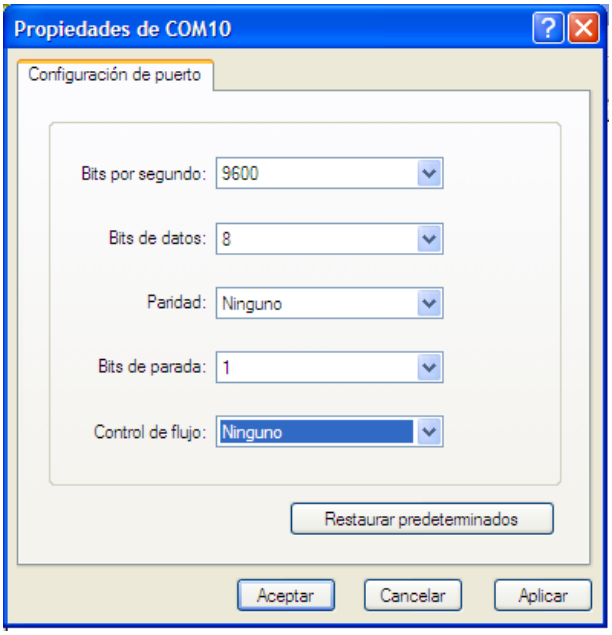
A continuación se muestra un ejemplo de cómo recibir los datos vía XBee con HyperTerminal:

Paso uno: Crear una nueva conexión

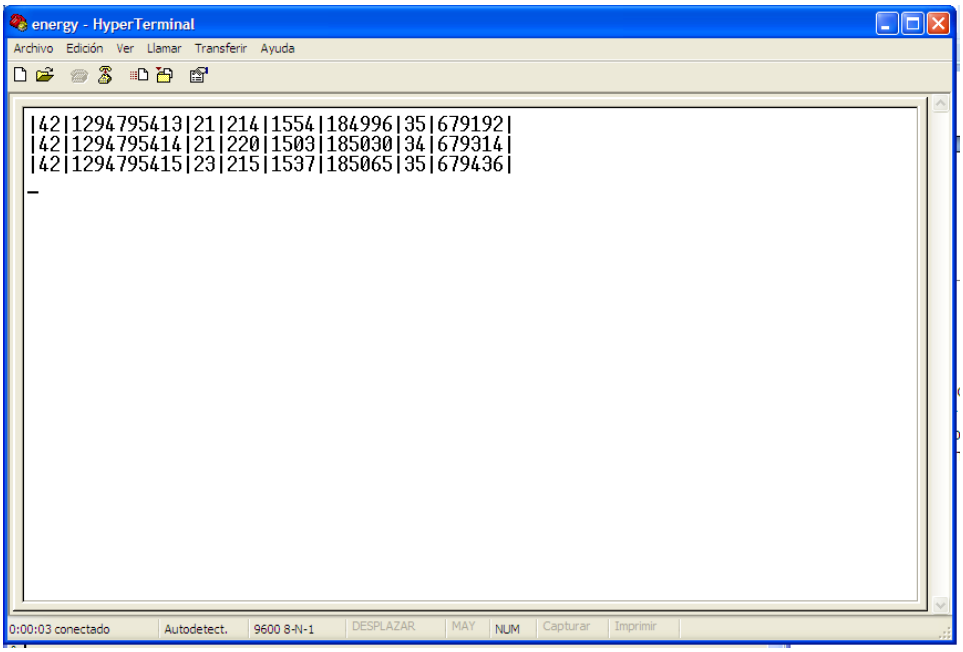
Paso dos: Escoger el puerto serie



Paso tres: Configurar la comunicación como se muestra en la imagen de a continuación



Paso cuatro: Los datos recibidos por el módulo XBee se mostrarán en la terminal



NOTAS

1. El módulo **NO** está diseñado para usarse directamente a 220V por los riesgos tanto para el equipo mismo como para las personas. Para proveer aislación galvánica deben utilizarse las sondas de voltaje y corriente adecuadas.

2. Al momento de validar alguna medición se recomienda descartar el primer valor obtenido por ser, generalmente, un resultado alterado, esto se debe a que el CI ADE7753 requiere un tiempo de estabilización.

3. Se recomienda sincronizar las mediciones con los cruces por cero de la señal de voltaje y tomar un promedio de muestras con el fin de tener resultados más estables.

4. Al momento de medir la energía activa acumulada es necesario que el consumo sea positivo, esto se puede llevar a cabo midiendo la energía activa acumulada en dos instantes de tiempo distintos y calculando su diferencia. En el caso de tener un consumo negativo (sabiendo previamente que es inconsistente este resultado), basta con invertir la polaridad de uno de los dos canales, ya sea el de corriente (canal1) o el de voltaje (canal 2).

MANTENIMIENTO

El equipo no requiere mayor mantenimiento salvo el recambio que debe hacerse cada cierto tiempo a la batería del reloj de tiempo real. Bajo condiciones normales la batería tendrá una duración de unas 235.000 horas, por lo que su reemplazo deberá efectuarse cada 5 años aproximadamente, o cuando se agote. Si el equipo se encuentra sin batería y es desenergizado, perderá su configuración horaria.

HISTORIA DEL DOCUMENTO

Revisión	Fecha	Editado por	Descripción/Cambios
1.0	10 de Febrero de 2011	S. Derteano	Versión inicial del documento
1.1	29 de Febrero 2012	R.Lobos	Incorporación de notas de uso y verificación de documentos y archivos de pruebas